

# appendix

## Answers to exercises

---

### Chapter 2

*Can you tell if a hash function provides hiding and binding if used as a commitment scheme?*

A hash function is *hiding* thanks to the pre-image resistance property; that is, if your input is random enough so that no one can guess it. To fix that, you can generate a random number and hash it with your input, and later, you can reveal both your input and the random number to *open* your commitment. A hash function is *binding* thanks to the second pre-image resistance property.

*By the way, there is no way this string represents 256 bits (32 bytes), right? How is this secure then?*

We don't care about collision resistance. We only care about second pre-image resistance. Thus, we can truncate the digest to reduce its size.

*Can you guess how the Dread Pirate Roberts (the pseudonym of Silk Road's webmaster) managed to obtain a hash that contains the name of the website?*

Dread Pirate Roberts created a lot of keys until one ended up hashing to that cool base32 representation. Facebook did the same and is accessible from facebookcore-wwi.onion (<https://facebook.com/notes/protect-the-graph/making-connections-to-facebook-more-secure/1526085754298237>). These are called *vanity addresses*.

## Chapter 3

*Can you figure out how a variable-length counter could possibly allow an attacker to forge an authentication tag?*

By observing the following message, where `||` represents string concatenation, `MAC(k, "1" || "1 is my favorite number")`, an attacker can forge a valid authentication tag for the eleventh message, `MAC(k, "11" || " is my favorite number")`.

*Caution: not all MACs are PRFs. Can you see why?*

Imagine that the following function is a valid MAC and PRF: `MAC(key, input)`, then is the following function a valid MAC? `NEW_MAC = MAC(key, input) || 0x01`? Is it a valid PRF? It is a valid MAC as it prevents forgery, but it's not a valid PRF, as you can easily distinguish the output from a totally random string (because the last byte is always set to 1).

## Chapter 6

*Using the same shared secret with everyone would be very bad; can you see why?*

If I can encrypt messages to you with this shared secret, I can also decrypt messages from other people.

*Do you see why you can't use the key exchange output right away?*

Remember what you've learned in chapter 5 on key exchanges. In (FF)DH, calculations happen modulo a large prime number  $p$ . Let's take a small prime number as example, 65,537. In hexadecimal, our  $p$  is written as `0x010001`, and in binary, it is written as `0000 0001 0000 0000 0000 0001`. In binary, notice the zeros preceding the first one because we represent our number in bytes (multiple of 8 bits).

If you understand modular arithmetic, you know that numbers modulo this prime  $p$  will never be larger, meaning that the first 7 bits will always be set to 0. In addition, the eighth bit will most often be set to 0 rather than 1. This is not *uniformly random*. Ideally, every bit should have the same probability of being set to 1 or to 0.

## Chapter 7

*As you saw in chapter 3, authentication tags produced by MACs must be verified in constant time to avoid timing attacks. Do you think we need to do the same for verifying signatures?*

No. This is because the verification of an authentication tag involves a secret key. Verifying a signature only involves a public key and, thus, does not need to be verified in constant time.

## Chapter 8

*Imagine for a minute that mixing different sources of entropy was done by simply XORing them together. Can you see how this might fail to be contributory?*

A backdoored source of entropy could set its output as the XOR of all the other sources of entropy, effectively canceling all entropy to 0.

*Signature schemes like BLS (mentioned in figure 8.5 and in chapter 7) produce unique signatures, but this is not true for ECDSA and EdDSA. Do you see why?*

In ECDSA, the signer can choose different nonces to produce different signatures for the same key pair and message. While EdDSA is a signature algorithm that deterministically derives the nonce based on the message to be signed, this does not mean that the signer cannot use any nonce if they so choose.

## Chapter 9

*A compromise of the server's private key at some point in time would be devastating as MITM attackers would then be able to decrypt all previously recorded conversations. Do you understand how this can happen?*

The attacker would then be able to rewind history and impersonate the server at the time the handshake was performed. Indeed, the attacker now has the server's private key. All the other information to perform the key exchange and derive the posthandshake symmetric keys is public.

*The values `signatureAlgorithm` and `signatureValue` are not contained in the actual certificate, `tbsCertificate`. Do you know why?*

The Certificate Authority (CA) needs to sign the certificate, which leads to a paradox: the signature cannot be part of the signature itself. The CA must, thus, append the signature to the certificate. Other standards and protocols might use different techniques. For example, you could include the signature as part of `tbsCertificate` and pretend that it is made of all 0s when you sign or verify the certificate.

## Chapter 10

*Do you know why the email's content is compressed before it is encrypted and not after?*

A ciphertext is indistinguishable from a random string according to the definition of a cipher. Due to this, compression algorithms are incapable of finding patterns to efficiently compress encrypted data. For this reason, compression is always applied before encryption.

*Can you think of an unambiguous way of signing a message?*

One line: authenticate the context. A way to do this is to include both the sender and the recipient's names and their public keys in the signature and then encrypt that.

## **Chapter 11**

*Sometimes applications attempt to fix the issue of the server learning about the user passwords at registration by having the client hash (perhaps with a password hash) the password before sending it to the server. Can you determine if this really works?*

Client-side hashing alone does not work as the infamous pass-the-hash attack showed ([https://en.wikipedia.org/wiki/Pass\\_the\\_hash](https://en.wikipedia.org/wiki/Pass_the_hash)); if the server stores Alice's hashed password directly, then anyone who steals it can also use it as a password to authenticate as Alice. Some applications perform both client-side hashing and server-side hashing, which, in this case, can perhaps prevent an active attacker from knowing the original password (although an active attacker might be able to disable client-side hashing by updating the code of the client application).

*Imagine a protocol where you have to enter the correct 4-digit PIN to securely connect to a device. What are the chances to pick a correct PIN by just guessing?*

That's 1 out of 10,000 chances to correctly guess something. You'd be happy if you were playing Lotto with these odds.

## Numerics

0-RTT (zero round trip time) 194  
3DES (triple-DES) symmetric key 281  
51% attack 265

## A

absorbing 41  
abstracting cryptography 17–18  
additive groups 99  
AddRoundKey function 69  
AEAD (authenticated encryption with associated data) 75–76  
AES (Advanced Encryption Standard) block cipher 66–70  
    amount of security provided by 67  
    interface of 67–68  
    internals of 68–70  
AES-CBC-HMAC construction 73–74  
AES-GCM AEAD 76–80  
AES-NI (AES New Instructions) 70  
append-only ledger 259  
applied cryptography 19  
Argon2 45  
arithmetic circuits 324, 338–339  
ASLR (address space layout randomization) 155  
ASN.1 (Abstract Syntax Notation One) 190  
associativity 92  
asymmetric cryptographic primitive 113  
asymmetric cryptography 10–16, 87, 303  
    digital signatures 15–16  
    key exchanges 10–12  
    vs. symmetric 13  
asymmetric encryption  
    in practice 111–117

    overview 110–111  
    with RSA 117–126  
        why not to use RSA PKCS#1 v1.5 121–123  
        with RSA-OAEP 123–126  
asymmetric keys 239–241  
    asymmetric password-authenticated key exchange 232–236  
    OPAQUE 234–236  
    OPRFs (oblivious pseudorandom functions) 233–234  
    mutual authentication in key exchanges 239–240  
    post-handshake user authentication with FIDO2 240–241  
asynchronous network 270  
attestation 289  
authenticated channels 244  
authenticated encryption 84  
    AEAD (authenticated encryption with associated data) 75–76  
    AES (Advanced Encryption Standard) block cipher 66–70  
        amount of security provided by 67  
        interface of 67–68  
        internals of 68–70  
    AES-CBC-HMAC construction 73–74  
    AES-GCM AEAD 76–80  
    CBC (cipher block chaining) mode of operation 70–73  
    ChaCha20-Poly1305 81–84  
    ciphers, overview of 65–66  
    encrypted penguin illustration 70–73  
authenticated key exchanges 89, 132–133  
authenticated strings 245  
authentication 182, 186  
authentication property 18

authentication tags 50  
     forgery of 53  
     lengths of 53–54  
     verifying in constant time 55–57  
 avalanche effect 70

## B

backdoors 20  
 backward secrecy 157  
 base blinding 294  
 BCryptGenRandom system call 159  
 BEAST (Browser Exploit Against SSL/TLS)  
     attack 73  
 Bell's theorem 301  
 beyond birthday-bound security 80  
 BFT (Byzantine fault-tolerant) consensus  
     algorithms 252–257  
     decentralization 254–255  
     distributed protocols 252–254  
     permissionless and censorship-resistant  
       networks 255–257  
 bilinear pairings 336–337  
 binding 32  
 birthday paradox 31  
 bit attack 42  
 Bitcoin 257–267  
     mining 259–265  
     reducing block's size by using Merkle trees 265–267  
     user balances and transactions 257–259  
 bitflip attacks 85  
 bit security 67  
 BitTorrent 32–33  
 blinding 294–295  
 blockchain 261  
 blockchain explorers 259  
 blockchain size 268  
 block cipher 36, 68  
 block function 81  
 blocks, Bitcoin 260  
 bootstrapping 328  
 boring cryptography 344  
 broken cryptographic algorithm 8  
 buffer overread 196

## C

CA/Browser Forum (Certification Authority  
     Browser Forum) 187  
 cache attacks 292  
 CAs (certificate authorities) 187, 201, 284  
 CBC (cipher block chaining) 71  
 CCA2 (adaptive chosen ciphertext attack)  
     122

CDNs (Content Delivery Networks) 32  
 certificate chain 188  
 Certificate message 239  
 certificate monitoring 195  
 CertificateRequest message 239  
 certificate revocation 195  
 certificates 188, 239  
 Certificate Transparency 197  
 CertificateVerify message 188, 239  
 ChaCha20-Poly1305 81–84  
 ChaCha20 stream cipher 81  
 chaos theory 153  
 checksum 26  
 ciphers, 65–66  
 ciphertext 65  
 ck (chaining key) 199  
 classifying cryptography 17–18  
 cleaning and mixing 159  
 clients 179  
 client-side fanout 223  
 closure 91  
 coinbase 262  
 collision resistance 29  
 collisions 53  
 commitments 32  
 commitment scheme 32  
 commit rule 273  
 committed changes to database 254  
 complexity of cryptography 24, 344  
 compression function 36  
 compromises 279  
 confidential computing 289  
 confidentiality 18, 268  
 confirmation blocks 264  
 consensus algorithms 253  
 constant-time programming 293–294  
 content integrity 33  
 contiguous rounds 273  
 contributory behavior 107  
 cookies 49, 58  
 correlation 301  
 counters 79  
 CPace (Composable Password Authenticated Con-  
     nection Establishment) 233, 245  
 CRLs (Certificate Revocation Lists) 195  
 CRS (common reference string) 334  
 cryptanalysts 8  
 cryptocurrency  
     BFT (Byzantine fault-tolerant) consensus  
       algorithms 252–257  
       decentralization 254–255  
       distributed protocols 252–254  
       permissionless and censorship-resistant  
       networks 255–257  
     Bitcoin 257–267  
       mining BTCs 259–262  
       mining conflicts 263–265

- cryptocurrency (*continued*)
  - reducing block's size by using Merkle trees 265–267
  - user balances and transactions 257–259
- DiemBFT 269–275
  - dishonesty toleration 270–271
  - rules of voting 271–272
  - safety and liveness 269–270
  - safety of 273–275
  - transactions, when considered finalized 273
- issues researched 267–269
  - blockchain size 268
  - confidentiality 268
  - energy efficiency 268–269
  - latency 267–268
  - volatility 267
- cryptographers 8
- cryptographic library 76
- cryptography
  - asymmetric 10–16
    - digital signatures 15–16
    - key exchanges 10–12
    - vs. symmetric 13
  - classifying and abstracting 17–18
  - complexity of 24
  - Kerckhoff's principle 7–9
  - practical 19–23
  - security protocols and 4–5
  - symmetric 5–6
  - theoretical vs. real-world 18–19
  - where and when fails
    - cryptography not an island 352
    - finding right cryptographic primitive or protocol is a boring job 344–345
    - good libraries 348–349
    - misusing cryptography 349–350
    - polite standards and formal verification 345–348
    - responsibilities as cryptography practitioner 353–355
    - usable security 351
- cryptography agility 346
- CRYSTALS (Cryptographic Suite for Algebraic Lattices) 314
- cSHAKE (customizable SHAKE) 42–44, 59
- CSPRNGs (cryptographically secure PRNGs) 155
- CTAP (Client to Authenticator Protocol) 241
- CTR (Counter) mode 76
- customization strings 42, 148
- CVP (closest vector problem) 313
- cyclic subgroups 94
- dd command-line tool 159
- decentralized randomness beacons 164
- decentralized trust 169–172, 208, 255
- decryption algorithm 65
- defense in depth 45, 279
- DEK (data-encryption key) 287
- DEM (data encapsulation mechanism) 116
- DER (Distinguished Encoding Rules) 191
- DES (Data Encryption Standard) 66, 281
- deserializing input 44
- deterministic consensus protocol 270
- deterministic ECDSA 144
- deterministic operations 68
- deterministic output 27
- DFA (differential fault analysis) 282
- DH (Diffie-Hellman) key exchange 91–98
  - discrete logarithm problem 95–97
  - group theory 91–95
  - standards 97–98
- DH (Diffie-Hellman) key pairs 215
- DH (Diffie-Hellman) ratchet 212, 220
- DiemBFT 269–275
  - dishonesty toleration 270–271
  - rules of voting 271–272
  - safety and liveness 269–270
  - safety of 273–275
  - transactions, when considered finalized 273
- digests 26, 37
- digital currency 251
- digital signatures 15–16, 90
- Dilithium signature scheme 316–318
- discrete logarithm problem 95–97
- disk encryption 85
- distributed database 252
- distributed systems 252
- DKG (distributed key generation) 171
- domain separation 42, 167
- DOS (denial of service) attacks 58
- dot product 313
- Double Ratchet protocol 218–222
- DPA (differential power analysis) 282, 291
- drand 164
- DRBGs (deterministic random bit generators) 155
- DRM (digital rights management) 280
- DSA (Digital Signature Algorithm) 22, 139
- DSKS (duplicate signature key selection) 149
- DUF (difference unpredictable function) 78

## D

- database encryption 85–86
- Davies–Meyer method 36

## E

- early boot entropy 162
- early data 194
- ECB (electronic codebook) 71

ECDH (Elliptic Curve Diffie-Hellman) key  
     exchange 97–105, 113, 125  
     elliptic curve overview 98–101  
     how works 102–103  
     standards for 103–105  
 ECDSA (Elliptic Curve Digital Signature  
     Algorithm) 139, 143–145, 257  
 ECIES (Elliptic Curve Integrated Encryption  
     Scheme) 116, 126–128  
 Ed25519 algorithm 146  
 Ed25519ctx algorithm 148  
 Ed25519ph algorithm 148  
 EdDSA (Edwards-curve Digital Signature  
     Algorithm) 139, 145–149  
 eigenvalue 330  
 eigenvector 330  
 encrypted email 110, 205–211  
     GPG 205–207  
     key discovery 208–210  
     PGP 205–207, 210–211  
     scaling trust between users with web of trust 208  
 encrypted penguin illustration 70–73  
 encryption algorithm 65  
 Encrypt-then-MAC 74  
 end-to-end encryption  
     encrypted email, failure of 205–211  
         GPG 205–207  
         key discovery 208–210  
         PGP 205–207, 210–211  
         scaling trust between users with web of  
         trust 208  
     reasons for using 202–203  
 Signal app 211–222  
     Double Ratchet protocol 218–222  
     more user-friendly than WOT 212–215  
     X3DH 215–217  
     state of 222–224  
 energy efficiency 268–269  
 ephemeral keys 144, 185  
 error range 314  
 Euclidian division 93  
 EUF-CMA model 149  
 eventual consistency 253  
 exfiltration attacks 209

## F

factorization problem 120  
 fault attacks 295–296  
 FBE (file-based encryption) 286  
 FDE (full-disk encryption) 286  
 FDH (Full Domain Hash) 143  
 federated protocols 212  
 fee field 44

FFDH (Finite Field Diffie-Hellman) 94  
 FHE (fully homomorphic encryption) 326–332  
     bootstrapping as key to fully homomorphic  
     encryption 328–330  
     FHE scheme based on learning with errors  
     problem 330–332  
     homomorphic encryption with RSA  
     encryption 327  
     types of homomorphic encryption 327–328  
     where used 332  
 FIB (focused ion beam) 279  
 FIDO2 (Fast IDentity Online 2) 240–241  
 fingerprints 213, 246  
 Finished message 189  
 finite field 94  
 FIPS (Federal Information Processing  
     Standards) 20, 67  
 fixed-sized compression function 37  
 forgery of authentication tags 53  
 forking processes 161  
 forks 254  
 formal verification 346  
 forward secrecy 156, 185, 207  
 FTS (few-time signatures) 308, 310  
 fully homomorphic encryptions 328  
 future secrecy 157

## G

Galois/Counter Mode 76  
 garbled circuits 326  
 Gaussian elimination algorithm 313  
 GDPR (General Data Protection Regulation) 203  
 general-purpose ZKPs (zero-knowledge proofs) 322  
 generators 94  
 getrandom system call 159  
 GPG (GNU Privacy Guard) 205–207  
 group messaging 223  
 group theory 91–95  
 Grover and Shor's algorithms 303–304

## H

handshake patterns 198  
 handshake phase 181  
 hardware attacks 279  
 hardware authenticators 240  
 hardware cryptography  
     choosing 289–290  
     HSMs (hardware security modules) 283–285  
     leakage-resilient cryptography 291–296  
         constant-time programming 293–294  
         fault attacks 295–296  
         masking and blinding 294–295  
     modern cryptography attacker model 278–279  
     smart cards and secure elements 281–283



hardware cryptography (*continued*)  
   TEE (trusted execution environment) 288–289  
   TPMs (Trusted Platform Modules) 285–288  
   white box cryptography 280  
 hash-based signatures 305–311  
   many-times signatures with XMSS and  
     SPHINCS+ 308–311  
   OTS (one-time signatures) with Lamport  
     signatures 305–306  
   WOTS (Winternitz one-time signatures) 307  
 hash functions 167  
   hashing passwords 44–46  
   in practice 31–33  
     BitTorrent 32–33  
     commitments 32  
     subresource integrity 32  
     Tor 33  
   overview 25–28  
   security considerations for 30–31  
   security properties of 28–30  
   standardized hash functions 34–44  
     avoiding ambiguous hashing with  
       TupleHash 43–44  
     SHA-2 hash function 35–38  
     SHA-3 hash function 38–41  
     SHAKE and cSHAKE 42–43  
 hash tables 58  
 HelloRetryRequest 184  
 heuristic-based constructions 17  
 hexadecimal 27  
 hidden variables 301  
 hiding 32  
 high availability 252  
 HKDF (HMAC-based key derivation function) 57,  
   164–168  
 HKDF-Expand function 165  
 HKDF-Extract function 165  
 HMAC (hash-based message authentication  
   code) 51, 58–59, 74  
 homomorphic commitments 335  
 homomorphic operations 336  
 HOTP (HMAC-based one-time password)  
   algorithm 237  
 HSMs (hardware security modules) 283–285, 289  
 HSTS (HTTP Strict Transport Security) 195  
 HTTP (Hypertext Transfer Protocol) 178  
 https 254  
 HTTPS (Hypertext Transfer Protocol Secure) 178  
 HVZK (honest verifier zero-knowledge) model 137  
 hybrid encryption 112–117  
 hypertrees 309

---

**I**

idempotent GET queries 194  
 identity authentication 227

identity element property 92  
 identity keys 215  
 IETF (Internet Engineering Task Force) 21, 178  
 imperfect cryptography 279  
 indistinguishable from random 156  
 info parameters 185  
 insecure channels 244  
 integrated security 288  
 interactive protocols 136  
 interactive ZKP systems 136  
 Internet protocol suite 178  
 invalid curve attack 106  
 invasive attacks 282  
 inverse element 92  
 iO (indistinguishability obfuscation) 280  
 IoT (Internet of Things) 282  
 IPSec 197  
 ISO/IEC 19790:2012 283  
 ISO 24759:2017 283  
 IVs (initialization vectors) 71

---

**J**

JavaCard 281

---

**K**

KDF (key derivation function) 125, 165, 217, 234  
 keccak-f permutation 38  
 KEK (key encryption key) 287  
 KEM (key encapsulation mechanism) 116  
 Kerckhoff's principle 7–9  
 key chains 169  
 key discovery 208–210  
 key distribution 10  
 key encapsulation 112–113  
 key exchanges 10–12, 112–113, 182  
   authenticated, as use case for signatures  
     132–133  
   DH (Diffie-Hellman) key exchange 91–98  
     discrete logarithm problem 95–97  
     group theory 91–95  
     standards 97–98  
   ECDH (Elliptic Curve Diffie-Hellman) key  
     exchange 98–105  
     elliptic curve overview 98–101  
     how works 102–103  
     standards for 103–105  
   forward-secure key exchanges and TLS 184–185  
   overview 88–91  
   security considerations 108  
   small subgroup attacks 105–108  
 key-extraction attack 291  
 key generation algorithm 110  
 key management 168  
 key pairs 11, 88

- key registries 208
- key revocation 169
- key rotation technique 169
- key-signing ceremony 212
- key stores 169
- keystreams 77
- key substitution attacks 149
- key wrapping 84–85
- KMAC 59
- KMS (Key Management Service) 350
- Koblitz curves 145
- Kyber key exchange 314–316

## L

---

- latency 267–268
- lattice-based cryptography 311–318
  - Dilithium signature scheme 316–318
  - Kyber key exchange 314–316
  - LWE (learning with errors) 313–314
  - overview 311–313
- layer 2 protocols 268
- leader election 254
- leakage-resilient cryptography 291–296
  - constant-time programming 293–294
  - fault attacks 295–296
  - masking and blinding 294–295
- length-extension attacks 58
- leveled homomorphic 328
- local attestation 289
- LWE (learning with errors) 313–314

## M

---

- MACs (message authentication codes) 74, 90, 131
  - example in code 51–52
  - in practice 58–59
    - HMAC 58–59
    - KMAC 59
  - in real world 57–58
    - deriving keys 57
    - hash tables 58
    - integrity of cookies 58
    - message authentication 57
  - security properties of 52–57
    - forgery of authentication tag 53
    - lengths of authentication tag 53–54
    - replay attacks 54–55
    - verifying authentication tags in constant time 55–57
  - SHA-2 and length-extension attacks 60–62
  - stateless cookies and 48–51
- MA-DH (Manually Authenticated Diffie-Hellman) 247
- malleable ciphertexts 122
- many time MAC 79

- many-times signatures, with XMSS and SPHINCS+ 308–311
- masking 294–295
- math-based constructions 17
- medium-term public keys 215
- membership proofs 266
- memory hard 46
- Merkle–Damgård construction 36
- Merkle trees 265–267, 308
- message/payload authentication 227
- message authentication 131
- message authenticity 227
- message key substitution attacks 149
- MFA (multi-factor authentication) 241
- MGF (mask generation function) 124
- million message attack 121
- minimum output size 31
- mining Bitcoin 259–265
- MITM (man-in-the-middle) attacks 12, 43, 89, 132, 278, 301
- MixColumns function 69
- MLS (Messaging Layer Security) 224
- modular arithmetic 92
- modular multiplication 92
- modulus 93
- Montgomery ladder’s algorithm 294
- MPC (multi-party computation), secure 322–326
  - general-purpose 324–325
  - PSI (private set intersection) 323–324
  - state of 326
- mTLS (mutually-authenticated TLS) 186
- multiplicative groups 91
- multi-signature systems 171
- mutually authenticated connections 242
- mutually-authenticated key exchanges 90, 133

## N

---

- next-generation cryptography
  - FHE (fully homomorphic encryption) 326–332
    - bootstrapping as key to fully homomorphic encryption 328–330
  - FHE scheme based on learning with errors problem 330–332
  - homomorphic encryption with RSA encryption 327
  - types of homomorphic encryption 327–328
  - where used 332
- MPC (multi-party computation), secure 322–326
  - general-purpose 324–325
  - PSI (private set intersection) 323–324
  - state of 326
- ZKPs (zero-knowledge proofs), general-purpose 332–342
  - arithmetic circuits 338–339

next-generation cryptography (*continued*)  
 bilinear pairings to improve homomorphic  
 commitments 336–337  
 from programs to polynomials 338  
 homomorphic commitments to hide parts of  
 proof 336  
 polynomials 340–342  
 R1CS (rank-1 constraint system) 339  
 succinctness 337–338  
 zk-SNARKs 335–336  
 no-cloning theorem 301  
 Noise protocol framework 197–200  
 noise sources 158  
 non-authenticated encryption 84  
 nonce misuse-resistant authenticated encryption 85  
 nonces 144  
 nondeterministic padding 121  
 non-interactive key exchanges 215  
 non-invasive attacks 282  
 not-perfect replacement 239

## O

OAEP (Optimal Asymmetric Encryption  
 Padding) 123, 142  
 OCSP (Online Certificate Status Protocol) 196  
 OCSP stapling 196  
 OIDC (OpenID Connect) 232  
 one-time MAC 79  
 one-time prekeys 215  
 OPAQUE 234–236  
 OpenPGP 205  
 OPRFs (oblivious pseudorandom functions)  
 233–234, 323  
 orders 94  
 origin/entity/identity authentication 227  
 origin authentication 131  
 OTPs (one-time passwords) 236  
 OTR (Off-The-Record) communication 211  
 OTS (one-time signatures) with Lamport  
 signatures 305–306

## P

padding 37, 70  
 padding bytes 70  
 padding section 61  
 partially homomorphic 327  
 password hashes 45  
 password hashing algorithm 229  
 passwords, replacing 228–241  
 asymmetric keys 239–241  
 mutual authentication in key exchanges  
 239–240  
 post-handshake user authentication with  
 FIDO2 240–241

asymmetric password-authenticated key  
 exchange 232–236  
 OPAQUE 234–236  
 OPRFs (oblivious pseudorandom  
 functions) 233–234  
 SSO (single sign-on) and password  
 managers 231–232  
 symmetric keys 236–239  
 PBFT (Practical BFT) algorithm 255  
 PCS (post-compromise security) 157, 212  
 pending changes to database 254  
 permissionless networks 256  
 permutations 38, 68  
 PGP (Pretty Good Privacy) 205–207, 210–211  
 physical unclonable functions 287  
 PKCS (Public Key Cryptography Standards) 22  
 PKCS#7 padding 71  
 PKCS#11 (Public Key Cryptography Standard  
 11) 284  
 PKI (public key infrastructure) 228, 254  
 plaintext-awareness property 124  
 platform authenticators 241  
 PoA (Proof of authority) 255  
 polite cryptography 345  
 Poly1305 core function 83  
 polynomials 338, 340–342  
 poor man's authentication 85  
 PoS (Proof of stake) 255  
 post-handshake 181, 246  
 post-quantum cryptography  
 hash-based signatures 305–311  
 many-times signatures with XMSS and  
 SPHINCS+ 308–311  
 OTS (one-time signatures) with Lamport  
 signatures 305–306  
 WOTS (Winternitz one-time signatures) 307  
 lattice-based cryptography 311–318  
 Dilithium signature scheme 316–318  
 Kyber key exchange 314–316  
 LWE (learning with errors) 313–314  
 overview 311–313  
 overview 304  
 PoW (proof of work) 256  
 precomputation attacks 235  
 preferred rounds 271  
 pre-image resistance property 28  
 primitives 5  
 private exponent  $d$  118  
 private keys 10  
 private randomness 163  
 PRNGs (pseudorandom number generators)  
 155–159  
 proof of knowledge 134  
 pseudo-anonymity 268  
 pseudo-primes 92  
 PSI (private set intersection) 323–324  
 PSKs (pre-shared keys) 193

public key cryptography 10  
 public\_keyE1 248  
 public key infrastructures 133  
 public keys 11, 88  
 public randomness 163

## Q

QC (quorum certificate) 270  
 QKD (quantum key distribution) 301  
 QR (Quarter Round) function 82  
 QR codes 214  
 QRNGs (quantum random number generators) 300  
 quantum bit 300  
 quantum computers 299–304  
   history of 302–303  
   impact of Grover and Shor’s algorithms on cryptography 303–304  
   overview 299–302  
 quantum entanglement 301  
 quantum gates 302  
 quantum mechanics 299  
 quantum-resistant algorithms 298  
 quantum-resistant cryptography 304  
 quantum superposition 300

## R

RICS (rank-1 constraint system) 339  
 randomness  
   decentralizing trust with threshold cryptography 169–172  
   generation of 161–163  
   key derivation with HKDF 164–168  
   managing keys and secrets 168–169  
   obtaining in practice 158–160  
   overview 153–155  
   PRNG (pseudorandom number generator) for slow randomness 155–158  
   public 163–164  
   security considerations 161–163  
 randomness extractors 155  
 random nonces 79  
 random oracle model 30  
 RDRAND 162  
 real-world cryptography 19  
 record padding 194  
 rekeying 80  
 related outputs 167  
 remainder of number 93  
 remote attestation 289  
 replay attacks 54–55  
 rewards, in Bitcoin mining 262  
 RFCs (Request For Comments) 22, 178  
 ROM (read-only memory) 287

root of trust 204, 287  
 round function 69  
 round keys 69  
 RSA 13, 327  
   asymmetric encryption with 117–126  
   RSA-OAEP 123–126  
   RSA PKCS#1 v1.5 121–123, 139–142  
   RSA-PSS 142–143  
 RSA-OAEP 117

## S

S/MIME (Secure/Multipurpose Internet Mail Extensions) 209  
 saltpack 210  
 salts 45, 165  
 SAML (Security Assertion Markup Language 2.0) 232  
 SAS (short authenticated strings) 246  
 scalar blinding 295  
 scalar multiplication 293  
 scaling to groups of larger membership 224  
 Schnorr identification protocol 134, 136–137  
 Schnorr signature scheme 138  
 Schrödinger’s cat experiment 300  
 Schwartz-Zippel lemma 337  
 searchable encryption 86  
 second pre-image resistance 26  
 secret keys 6, 65  
 secrets, managing 168–169  
 secret sharing 169  
 secure boot 287  
 secure cryptographic algorithms 7  
 secure elements 282  
 secure messaging 211  
 secure transport  
   Noise protocol framework 197–200  
   SSL secure transport protocol 178–179  
   state of encrypted web today 194–197  
   TLS secure transport protocol 181–194  
     authentication and web public key infrastructure 186–189  
     authentication via X.509 certificates 190–193  
     avoiding key exchanges 193–194  
     forward-secure key exchanges and 184–185  
     from SSL to 178–179  
     how TLS 1.3 encrypts application data 194  
     negotiation in 182–184  
     pre-shared keys and session resumption in 193–194  
     using in practice 179–180  
 security claims 17  
 security protocols 4–5  
 security through obfuscation 280  
 security through obscurity 8, 280  
 seeds 155, 218

- semi-invasive attacks 282
- sending chain key 219
- sending key 219
- serializing inputs 44
- server parameters 186
- servers 179
- server-side fanout 224
- session resumption 182
- session tickets 193
- SHA-1 (Secure Hash Algorithm 1) 35
- SHA-2 hash function 35–38
- SHA-3 hash function 38–41
- SHAKE 42–43
- ShiftRows function 69
- Shor and Grover’s algorithms 303–304
- short Weierstrass equation 98
- Sigma protocols 136
- Signal app 211–222
  - Double Ratchet protocol 218–222
  - more user-friendly than WOT 212–215
  - X3DH 215–217
- signature aggregation 171
- signatureAlgorithm 190
- signature forgeries 141, 307
- signatures
  - malleability of 150
  - origin of 134–138
  - overview 130–133
  - public key infrastructures and 133
  - signature algorithms 138–149
    - ECDSA (Elliptic Curve Digital Signature Algorithm) 143–145
    - EdDSA (Edwards-curve Digital Signature Algorithm) 145–149
    - RSA PKCS#1 v1.5 standard 139–142
    - RSA-PSS standard 142–143
  - signing and verifying in practice 131
  - substitution attacks on 149
  - use case for 132–133
  - ZKPs (zero-knowledge proofs) and 134–138
    - Schnorr identification protocol 134–137
    - signatures as non-interactive zero-knowledge proofs 137–138
- signature schemes 130
- signatureValue 190
- signed prekeys 215
- signing key 130
- signing signatures 131
- single point of failure 171, 252
- single-use public keys 215
- SipHash 58
- SIV (synthetic initialization vector) 85
- skimmers 278
- small subgroup attacks 105–108
- smart cards and secure elements 281–283
- SMTP (Simple Mail Transfer Protocol) 209
- software attacks 279
- somewhat homomorphic 327
- SPA (simple power analysis) attack 291
- SPHINCS+ signature scheme 308, 310–311
- sponge construction 38
- square and multiply 96
- squeezing 41
- SRP (Secure Remote Password) 232
- SSH (Secure Shell) protocol 197, 213, 240
- SSL (Secure Sockets Layer) protocol 178–179
- SSO (single sign-on) 231–232
- SSS (Shamir’s Secret Sharing) 170
- stablecoin 267
- standardization threshold 333
- standardized hash functions 34–44
  - avoiding ambiguous hashing with TupleHash 43–44
  - SHA-2 hash function 35–38
  - SHA-3 hash function 38–41
  - SHAKE and cSHAKE 42–43
- state 40, 68
- stateless cookies 48–51
- static keys 199
- SubBytes function 69
- subgroups 94, 117
- subresource integrity 32
- substitution attacks 149
- sum 26
- SVP (shortest vector problem) 312
- sybil attacks 263
- symmetric cryptographic primitives 113
- symmetric cryptography 5–6, 303
- symmetric encryption 5, 84–86
  - database encryption 85–86
  - disk encryption 85
  - key wrapping 84–85
  - nonce misuse-resistant authenticated encryption 85
  - vs. asymmetric 13
- symmetric keys 236–239
- symmetric ratchet 219

---

**T**

- Tamarin protocol prover 347
- tbsCertificate field 190
- TCG (Trusted® Computing Group) 285
- TCP (Transmission Control Protocol) 178
- TCP frames 178
- TDE (transparent data encryption) 86
- TEEs (trusted execution environments) 288–290
- theoretical cryptography 18, 345
- threat model 203
- threshold cryptography 169–172, 322
- threshold distributed keys 164
- timing attacks 55
- tls13 c hs traffic 185

tls13 s ap traffic 185  
 TLS secure transport protocol 181–194  
   authentication and web public key  
     infrastructure 186–189  
   authentication via X.509 certificates 190–193  
   avoiding key exchanges 193–194  
   forward-secure key exchanges and 184–185  
   from SSL to 178–179  
   how TLS 1.3 encrypts application data 194  
   negotiation in 182–184  
   pre-shared keys and session resumption in  
     193–194  
   using in practice 179–180  
 TOFU (trust on first use) 212, 227  
 tooling 349  
 Tor 33  
 total breaks 70, 144  
 TOTP (time-based one-time password)  
   algorithm 237  
 TPMs (Trusted Platform Modules) 285–288  
 transcript consistency 224  
 TRNGs (true random number generators) 155  
 truncating digests 31  
 trust  
   decentralizing with threshold cryptography  
     169–172  
   scaling trust between users with web of trust 208  
 TupleHash 43–44  
 twisted Edwards curves 146

## U

unauthenticated key exchanges 89, 185  
 unencrypted protocols 205  
 uniform distribution 153  
 unique IV 73  
 unpredictable IV 73  
 unpredictable secrets 158  
 user-aided authentication 227  
 user authentication 227  
   passwords, replacing 228–241  
     asymmetric keys 239–241  
     asymmetric password-authenticated key  
       exchange 232–236  
     SSO (single sign-on) and password  
       managers 231–232  
     symmetric keys 236–239  
   user-aided authentication 242–248  
     pre-shared keys 244–245  
     SAS (short authenticated string) 246–248  
     symmetric password-authenticated key  
       exchanges with CPace 245–246  
 userland PRNGs 161  
 UTXOs (Unspent Transaction Outputs) 258

## V

vector space 311  
 verifying key 130  
 verifying signatures 131  
 VMs (virtual machines) 162  
 volatility 267  
 VRFs (verifiable random functions) 163

## W

WebAuthn (Web Authentication) 241  
 web PKI (web public key infrastructure) 133, 187  
 white box cryptography 280  
 wide-block ciphers 85  
 WOT (web of trust) 208, 227  
 WOTS (Winternitz one-time signatures) 307  
 WPA (Wi-Fi Protected Access) 197

## X

X25519 key exchange algorithm 91  
 X3DH (Extended Triple Diffie-Hellman) 212,  
   215–217  
 X.509 certificates 190–193  
 XML (Extensible Markup Language) 232  
 XMPP (Extensible Messaging and Presence  
   Protocol) 211  
 XMSS (extended Merkle signature scheme)  
   308–311  
 XOFS (extendable output functions) 42–43, 124,  
   168

## Z

Zcash 268  
 zeroization 283  
 ZKPs (zero-knowledge proofs) 134–138, 268,  
   332–342  
   arithmetic circuits 338–339  
   bilinear pairings to improve homomorphic  
     commitments 336–337  
   homomorphic commitments to hide parts of  
     proof 336  
   polynomials 338, 340–342  
   RICs (rank-1 constraint system) 339  
   Schnorr identification protocol 134–137  
   signatures as non-interactive zero-knowledge  
     proofs 137–138  
   succinctness 337–338  
 zk-SNARKs (Zero-Knowledge Succinct Non-  
   Interactive Argument of Knowledge)  
   334–336

# Real-World Cryptography

David Wong

Cryptography is the essential foundation of IT security. To stay ahead of the bad actors attacking your systems, you need to understand the tools, frameworks, and protocols that protect your networks and applications. This book introduces authentication, encryption, signatures, secret-keeping, and other cryptography concepts in plain language and beautiful illustrations.

**Real-World Cryptography** teaches practical techniques for day-to-day work as a developer, sysadmin, or security practitioner. There's no complex math or jargon: Modern cryptography methods are explored through clever graphics and real-world use-cases. You'll learn building blocks like hash functions and signatures; cryptographic protocols like HTTPS and secure messaging; and cutting-edge advances like post-quantum cryptography and cryptocurrencies. This book is a joy to read—and it might just save your bacon the next time you're targeted by an adversary after your data.

## What's Inside

- Implementing digital signatures and zero-knowledge proofs
- Specialized hardware for attacks and highly adversarial environments
- Identifying and fixing bad practices
- Choosing the right cryptographic tool for any problem

For cryptography beginners with no previous experience in the field.

**David Wong** is a cryptography engineer. He is an active contributor to internet standards including Transport Layer Security.

Register this print book to get free access to all ebook formats.  
Visit <https://www.manning.com/freebook>

“A staggeringly comprehensive review of the state of modern cryptography. Essential for anyone getting up to speed in information security.”

—Thomas Doylend  
Green Rocket Security

“Covers all the important parts of applied cryptography. A must read for every Java developer.”

—Harald Kuhn, TurnFriendly

“An in-depth introduction to cryptography, covering hot topics like blockchain and quantum computing.”

—Gábor László Hajba, ProLion

“A best-in-category book that takes you all the way from curious novice to confident practitioner.”

—William Rudenmalm  
Creandum



ISBN: 978-1-61729-671-0



9 781617 296710